

## **REMARKS**

Claims 1-3, 5-12, 31, and 35-79 are pending and stand rejected.

Claims 1, 31, 42, 48, 61, and 67 are amended herein.

Accordingly, claims 1-3, 5-12, 31, and 35-79 are pending upon entry of this amendment.

Claims 1-3, 10, 12, 35-37, 40-45, 49-53, 61-65, and 68-71 stand rejected under 35 U.S.C. §102(e) as being anticipated by Turski (U.S. Patent Application Publication 2004/0255301).

Claims 5-9, 11, 38, 39, 47, 54-60, 66, and 73-79 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Turski in view of Cotton (U.S. Patent 7,016,919). Claims 31, 48, and 67 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Turski in view of Schumacher (U.S. Patent 6,532,023).

Applicants respectfully traverse these rejections as applied to the amended claims.

Independent claim 1 as amended recites a method for capturing event data associated with a plurality of different types of articles generated by a plurality of different client applications:

- storing a plurality of different event schemas, each event schema associated with at least one of the types of articles;
- detecting an event, the event including a user interaction with an article; responsive to the event, determining an event schema associated with the type of the article; and
- storing event data identifying the event and the article using the determined event schema.

The claimed method enables the use of *different* event schemas to capture the various events that occur on a client device, where there are different types of articles with which a user interacts. The claimed method determines the appropriate one of the different event schemas associated with the type of the article with which the user interacts and stores event data identifying the event and the article using the determined event schema. This is beneficial for

several reasons. First, it enables capture of event data for an article using a specific event schema that is associated with the article type. Second, the capture of event data for an article using a specific event schema associated with the article type beneficially improves indexing, storing, and searching of the event data associated with various articles based on article type.

Turski does not disclose the claimed method. Turski discloses a single generic schema that associates a pair of objects with a third associating object using an association table 302 and an object table 304 (paragraphs [0035-0038], FIG. 3). Thus, Turski does not have a plurality of “*different event schemas*” for “*different types of articles*” associated with “*different client applications*” as claimed. Furthermore, Turski does not store data using a plurality of different event schemas for different article types. As a result, Turski does not afford the benefits that come from the ability to store, index, and search event data using different event schemas. Additionally, Turski’s single generic schema associates events in relationship to other events and teaches away from selecting a schema responsive to an event by stating that such “conventional out-of-context filing structures” make it difficult or impossible to find historical information (paragraph [0030]).

Independent claims 42 and 61 also recite “storing a plurality of different event schemas, each event schema associated with at least one of the types of articles; detecting an event, the event including a user interaction with an article”. Thus, Turski does not disclose the elements of these claims, and hence independent claims 1, 42, and 61 are allowable over Turski. The dependent claims incorporate the limitations of their respective base claims and are allowable for at least the same reasons.

With respect to claims 5-9, 11, 38, 39, 47, 54-60, 66, and 73-79, the combination of Turski and Cotton does not disclose or suggest the limitations of these claims, at least in part because Cotton does not remedy the deficiencies of Turski, as described above. Cotton discloses a web-based framework wherein events and associated timestamps and traceability reports are stored with the meta-data associated with application data, thereby creating a history of all events that have occurred with the data (col. 4, lines 34-37). At best then, Turski and Cotton merely teach a system for storing event objects in the object table of the single generic schema together with meta-data that contain timestamps and traceability reports. Turski and Cotton in combination do not provide different schemas for different types of articles associated with different client applications. Accordingly, Turski and Cotton do not disclose the elements of these claims.

With respect to amended claims 31, 48, and 67, the combination of Turski and Schumacher does not disclose or suggest the limitations of these claims, at least in part because Schumacher does not remedy the deficiencies of Turski, as described above. Schumacher discloses detecting an applet event and queuing the applet event in an automator queue (col. 6, lines 52-55). At best, Turski and Schumacher merely teach a system for storing, from an automator queue, event objects in the object table of the single generic schema. Turski and Schumacher in combination do not provide different schemas for different types of articles associated with different client applications. Accordingly, Turski and Schumacher do not store a plurality of different event schemas for different article types. Further, Turski and Schumacher do not place events in a queue in the format described by one of a plurality of different event schemas. Accordingly, Turski and Schumacher do not disclose the claimed invention.

The Examiner is encouraged to contact the undersigned attorney if it would be beneficial to further advance the prosecution of the application.

Respectfully submitted,  
MIHAI FLORIN IONESCU ET AL.

Dated: December 4, 2007 By: /Robert R. Sachs/

Robert R. Sachs, Reg. No. 42,120  
Attorney for Applicants  
Fenwick & West LLP  
Silicon Valley Center  
801 California Street  
Mountain View, CA 94041  
Tel.: (415) 875-2410  
Fax: (415) 281-1350  
Email: rsachs@fenwick.com